DOCUMENT RESUME

ED 427 727 IR 019 278

AUTHOR Newmarch, Jan
TITLE An Investigation into the Effect of Hypertext Structure on Student Use of Courseware.
PUB DATE 1998-11-00
NOTE 7p.; In: WebNet 98 World Conference of the WWW, Internet and Intranet Proceedings (3rd, Orlando, FL, November 7-12, 1998); see IR 019 231. Figure 4 may not reproduce clearly.
PUB TYPE Reports - Evaluative (142) -- Speeches/Meeting Papers (150)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Assisted Instruction; Computer Science Education; Costs; Course Content; *Courseware; Foreign Countries; Higher Education; *Hypermedia; *Material Development; Measurement Techniques; Navigation (Information Systems); Student Reaction; Use Studies; World Wide Web
IDENTIFIERS Computer Operating Systems; Links (Indexing); University of Canberra (Australia); Web Pages

ABSTRACT

The University of Canberra (Australia) has made courseware on the subject of computer operating systems available on the World Wide Web for a number of years, using a very simple hypertext format. Server logs have previously been analyzed to determine the amount of student usage. This paper reports on a restructuring of the courseware designed to produce increased use by means of a more sophisticated hypertext structure. The time costs in this restructuring, as well as the failure to increase usage, are reported. This has particular implications for educational institutions attempting to persuade instructors to place courseware on the Web. Five figures present the simple course structure, fine course structure, flexible course structure, monthly usage of old and new pages, and alternative routes through the courseware. (Author/AEF)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

An Investigation into the Effect of Hypertext Structure on Student Use of Courseware

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☐ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Jan Newmarch
Distributed Information Laboratory, University of Canberra, Australia
jan@ise.canberra.edu.au, http://pandonia.canberra.edu.au

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
G.H. Marks
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Abstract: *Courseware for a particular University subject has been available on the Web for a number of years using a very simple hypertext format. Server logs have previously been analysed to determine the amount of student usage. This paper reports on a restructuring of the courseware designed to produce increased use by means of a more sophisticated hypertext structure. The time costs in this restructuring as well as the failure to increase usage are reported. This has particular implications for educational institutions attempting to persuade instructors to place courseware on the Web.*

ED 427 727

1. Introduction

Generating course materials in a suitable form for students is a principal function of a instructor's job. The time spent in preparing this courseware, and the corresponding cost in the salary of the instructor and any support staff can vary greatly depending on the presentation forms chosen. This time - and increasingly for computer-based systems, also maintenance time - has to be factored in when preparing courseware.

Web courseware has traditionally (i.e. over the past few years) been viewed as "free" due to the enthusiasm of early adopters prepared to develop materials in their own time, and due to the lack of charges made to view the courseware. This is now changing: partly because an increasing number of Universities are restricting access to courseware, limiting it to their own revenue-raising students. In addition, University pressure for flexible delivery mechanisms is forcing more academics to place courseware on the Web even though they may feel it is an additional burden on their time.

The subject Operating Systems at the University of Canberra has been available in Web format since 1994 [Newmarch 1996]. This was done with very few additional overheads by converting from an existing electronic form, by placing each section of the course on the Web as a single document and adding very little hypertext structure. This simplified maintenance (as well as other things such as making printing simple!).

From a low use in 1994 because the Web was young, the courseware had quite high usage in 1995. In 1996 this fell off, possibly because students realised that the Web form used was very similar to the printed form and lacked many useful features of paper-based versions, such as being able to be taken to a lecture and annotated.

In 1997, a major restructuring of the courseware was undertaken. The main purpose of this was to add features that were not possible in a paper-based version, such as multiple guided paths through the courseware. The intent was to see if this would lead to increased use by students and increased satisfaction, but also to assess the extra time cost of preparation in a more complex form. The restructuring was performed by a single (knowledgeable) instructor - this is looking at the kind of work that may be performed by an individual or a small team rather than by a major University effort.

This paper tests the hypothesis that the navigational structure of Web courseware affects the amount of use students will make of the courseware. It also estimates the extra time cost that is involved in adding navigational aids to courseware.

2. Simple Structure

Courses at the University of Canberra in Information Technology areas are usually lecture-based. Students expect to receive instruction in one hour-sized chunks. Each lecture generally consists of a single major topic. Some topics may be carried on beyond this scale, but each lecture forms a relatively self-contained whole. For example, for the subject Operating Systems, lectures were: File Systems, Directories, Memory Management, Virtual Memory, etc. The simplest organisation is to prepare each lecture as a single HTML document.

IR019278

BEST COPY AVAILABLE

A central navigation page can list all of these lecture pages and any others of interest such as assignments and tutorial exercises. Links between lecture pages can be kept to a minimum. This gives a tree structure with one level in Figure 1.

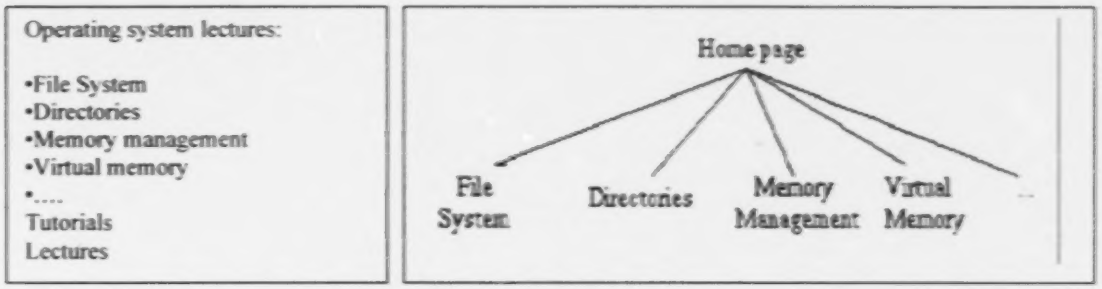

Figure 1: Simple course stucture

This provides about the minimal structure needed to make the material available on the Web. It has the advantages of being easy to maintain the hypertext structure (since there isn't much) and being easy to print. Keeping the content up to date is an easy editorial task as there are not many files to edit. Essentially, this type of Web structure is just a table of contents of a set of lectures.

The Operating System lecture notes were kept in this form from 1994 to 1996. Access patterns to this were reported in [Newmarch 1997]. In brief: usage was low in 1994 because the Web was young, was high in 1995, and fell for local students in 1996. I suspect this fall was caused by two main factors: the novelty value wore off, and the students realised that there were only a few advantages over a paper-based version. Another factor may have been speed: Netscape grew in size, and we switched from SunOS to Linux in our own computer laboratories with a less efficient version of NFS. Startup times for Netscape were in the minutes rather than seconds, leading to less students bothering with it when a paper version was handy. Students had the advantage of paper-based versions being easily available through the local bookshop, and were able to make judgements on use based on the availability of these alternative media.

3. Fine Structure

A lecture on a topic such as File Systems will have many internal components: the general services provided to applications by a file system; the Unix API for such service; the DOS API for the same set; implementation techniques such as inodes, and so on. Generally these are broken into further subtopics, until the level of individual ``slides" or other indivisible components is reached. For example, the ``table of contents" may be expanded out as in Figure 2.

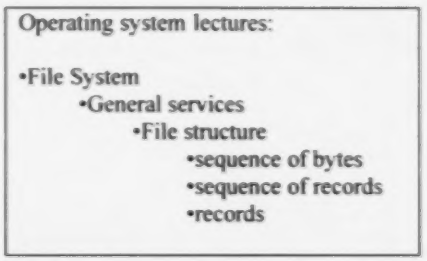

Figure 2: Fine course structure

Most linear courseware lends itself to a hierarchical structure that can be captured as a tree of hypertext links. Hypertext however encourages making alternative routes through courseware so that a student can choose the path they want to follow. These need not follow the hierarchical structure, but may cut across it. Such alternative routes must follow a structure of their own that is a valid structure for the couresware.

4. Intermediate Structure

The Operating Systems subject has a lecture-level structure, each lecture has a set of subtopics, and so on until we reach components that are indivisible. To produce alternative routes means linking the components in different ways.

Each lecture should have an overview, no matter what the courseware is about. This can be used as one alternative route, to give a set of overviews of the various topics discussed in the course Typically, each topic may also contain a summary, and these could be linked.

Further routes depend on the particular courseware, and how coherent it remains in moving from one topic to another. Some may have little coherence, some may have a lot. For this particular courseware it was possible to identify groupings of components, fairly similar to the subtopics of each lecture. Within each lecture there was usually some general theory, illustrations using the Unix Operating System and other illustrations using MSDOS or Windows 95.

These led to alternative routes through the courseware, as Overview, Generic, Unix and Win32. Some of the original material needed to be regrouped, but by and large it was fairly easy to place indivisible components into one of these four groups. In some cases a group really contains two or more subtopics, but these could be broken into separate parts without affecting the overall structure. Applying this to the courseware gave views such as the Lecture view, the Unix view and the Generic view in Figure 3.

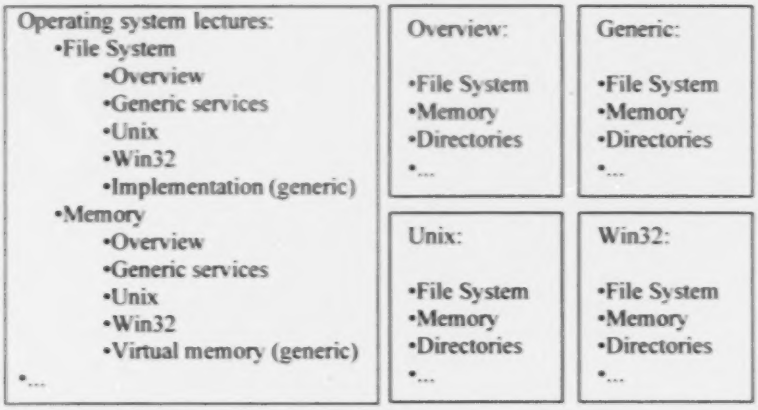

Figure 3: Flexible course structure

This structure was custom-designed for this subject. Other subjects may have greater or lesser success in finding such an intermediate structure, and the alternative navigation paths that this provides.

5. Time Costs

5.1 Simple structure

The simple structure is extremely low-cost. The original material for this was already in word-processor format, in fact as Interleaf documents. In 1994 there were no automatic converters to HTML. After a few experiments, it was decided to save the documents in plain ASCII text and add HTML markup by hand. This markup task was performed by a secretary in about a week for a complete semester subject. As an aside, two years later she expressed a desire to learn HTML, and couldn't even remember having marked up an entire subject earlier!

If the courseware is already in word-processor format, most likely there are now converters to HTML. This can reduce this initial markup even further. If the courseware has to be produced from scratch, the use of HTML-editors makes it as easy/hard as any other document formatting system such as WordPerfect.

Maintenance is also low-cost. The major task is text editing, with a few links to graphics files. If the courseware is relatively stable, this may take very little time.

5.2 Intermediate Structure

5.2.1 Granularity

Deciding the logical structure is a non-trivial operation. There will be a coarse breakdown in topics, and this was used in the simple structure. To produce a more fine-grained structure is far more complex: while it is easy to break any topic into smaller components, to break it into a set that is consistent with the breakup of all other topics is harder. An instructor will probably be aware of many possibilities for this finer structure, but it is necessary to decide between them and check that it will work adequately for all topics. Producing a coherent set of views for a course should have educational advantages.

Producing this structure is a significant effort, because it will affect both the work needed to create the courseware and the student perception of it. For Operating Systems, this task alone took nearly a week, and may not be possible for some courseware.

5.2.2 Links

The ability to allow multiple paths through a document set is a desirable feature of hypertext systems. This is intimately tied up with the finer structure of the courseware, as alternative navigation views will be alternative routes through this finer structure.

Ideally, the links need to be external to the documents. This allows a document to concern itself with content only, while the relation to other documents is handled elsewhere. However, this is not supported by HTML 3.2, as hyper links are encoded directly into documents. It is directly supported by hypertext systems such as Hyper-G (renamed HyperWave) [Maurer 1996], where the problems of embedded hyperlinks are described.

Many browsers support Frames, and this will become part of the HTML 4.0 standard. Frames provide a partial solution to this by allowing selection in one frame to cause navigation in another. Frames allow some navigation links to be handled externally to the documents. Using frames gives a physically more complex structure to the Web pages. For Operating Systems, about half of the pages are just support pages for the frames. Setting this up in a consistent manner across all pages took a week. This involved building a suitable directory structure, creating template files and also writing some CGI scripts to ensure that switching from one view to another updated all of the frames appropriately. Maintenance is also costly: to add a new element to a topic invoves updating four other pages; to add a new topic involves setting up a new directory with template files and updating six other pages.

This adds significant overheads to content preparation. The logical structure needs to be mapped onto the physical file structure. All navigation files need to be updated when changes are made. While parts of this can be automated, the failure of the topics to fit precisely into a fixed structure works against this. Different subjects will have different structures, so a general automated solution seems far away. I take about an hour to integrate a new topic into the existing ones, but then I know exactly the logical and physical structures I am dealing with. Other instructors may require the support of technical staff.

The cost of ongoing maintainence has not yet been measured as this is the first time it has been given in this form. However, fixing up ``bugs" - such as wrong links and incorrect titles in HTML documents - has cost many hours.

One unexpected cost arose in looking at search engines: most of the commercial search engines will not index through HTML frames [DeUlloa 1997], so all of the content was rendered invisible to them by using the frame solution adopted here. A week's work was involved in producing a non-frame version that would allow search engines to index the content. The revised pages still had to work with the current browsers that do understand frames, of course! Testing this was messy - it involved finding a browser that did not understand frames, and a version of Lynx on one of our older systems was finally used for this.

5.2.3 Navigation

In addition to logical navigation through courseware, students also expressed the need for position information - the added complexity made some people feel "lost" in the courseware. Position information can be given by consistent naming for both document title and first level header in the document that follows the logical structure of the courseware. This may produce fairly cumbersome names such as ``Operating Systems - File Systems - Generic Operations". Ensuring this is just part of the quality control of the courseware: something that just takes a few minutes, but has to be built into the process of creating documents.

For this subject, a map of the subject contents was generated by a CGI script, which located the user in the content element and allowed navigation to other elements. This map ignored things such as assignments, and tutorials and showed a simplified map of "real" content only. Building the CGI script for this took another week, which would not have been necessary in the simple navigation model.

6. Measures

6.1 Source Data

Surveys have been kept on the subject in all years. They show what surveys of other subjects show: approval of Web-based delivery [good questionnaire]. What they do not show is the use students actually make of the material, and this requires more objective measures. This is not to dismiss surveys, but to note that positive survey results are difficult to interpret if they form the only measure of the courseware.

Web servers can be configured to keep logs of all accesses made from them. Server log statisics need to be interpreted with care: for example, they do not measure the number of requests made, but only the number of requests that make it to the server [Noonan 1998] [Linder 1998] [Goldberg 1998]. The rest may be delivered from proxy caches along the way. Logs contain information about requests for gif images along with requests for documents. Generally, these will need to be filtered out before meaningful conclusions can be drawn.

6.2 Accesses to Courseware

Access patterns to the original lecture notes were reported in detail in [Newmarch 1997]. Tables for the new version are given in [Newmarch 1998]. Comparing the ``best" previous year of 1995 to the restructured pages of 1997 gives the graph of Figure 4.

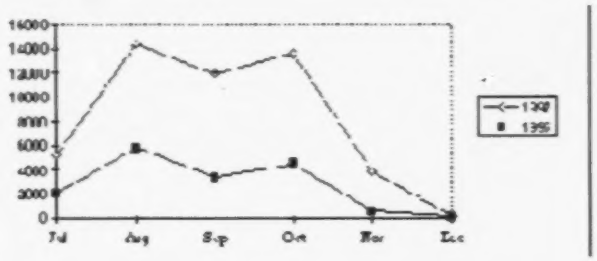

Figure 4: Monthly usage of old and new pages

The graphs deal with about 120 students in 1995 but only 90 in 1997. On the face of it, an improvement in usage has taken place. However, in 1997 the page structure was more complex, with about half of the pages being purely for navigation. In addition, each of the 1995 pages was split into four or more pages for 1997. Taking these into account reduces the difference so that there is no noticeable change in usage.

A "control" group of 30 students continued to use the old version, as they were taught the older version of the course. Making the same adjustments for structure, the control group used the old pages with the same frequency as the new students.

6.4 Cookies

Cookies are typically used by server administrators to track the set of pages retrieved during a "session", so that routes through the pages can be traced reliably. Cookies were added to the server for this site in order to determine if the alternative navigation routes were actually used. If they were, it justifies the structure adopted. If not, it weakens one of the reasons for this structure.

In [Newmarch 1997] it was reported that over half of the accesses to the Web courseware only looked at one page. This pattern has continued unchanged in the revised courseware. In one time period, there were 15.000 cookies recorded. However, 8,000 of these only looked at one page. Of those who explored further, the results of Figure 5 were recorded. These figures show that use was made of the new structure - although the "lecture" route was preferred, about a quarter of that number also explored the ``Unix" and ``Win32" routes.

route	lectures	Overview	generic	Unix	Win32
number	2815	314	423	730	698

Figure 5: Alternative routes through courseware

7. Time Costs

A simple structure is easy to setup and low-cost to maintain. Conversion from a previous document format only took a week when few tools were available.

For this subject, devising and implementing the more flexible structures cost about two weeks. Bringing in a new topic added about an hour of work, so two topics per week over a semester added about thirty hours of work in addition to preparation of the material in the first place. Maintenance time rose sharply using the more complex structure. The structure was not quite regular enough for an automated maintenance system. Minor errors had to be fixed on a regular basis. Tools to establish link consistency on the site had to be learnt and used. Modifications to allow search engines to access the courseware were unexpected. A site map had to be built to aid those who found the new structure too complex.

In summary, the amount of extra time spent on the courseware to build and maintain the more complex structure added up to about six weeks of work. Some of this involved highly technical work, writing CGI scripts. This is a significant overhead for a one semester course taught by a single instructor. If the courseware remains stable and reduces to maintenance only this should drop to a few days per semester, but of course this is unlikely given the rapid evolution of the Web.

8. Conclusion

Interest by students in Web pages that added little to printed versions of the courseware flagged after a few years. To regain this, a more sophisticated hypertext structure was put in place to allow students to navigate more flexibly around the courseware. This feature could not be done with traditional paper-based courseware. This added significantly to overheads on course preparation and maintenance, more than could normally be expected of a professor without significant assistance.

Student response was highly favourable to the new course structure. Cookie logs showed that students did make use of this additional flexibility. However, the logs showed that overall it did not result in an observable increase in the use of the courseware.

This paper varied the parameter of Web courseware structure in order to test the hypothesis that the structure will affect the amount of use students will make of the courseware. The results do not confirm this hypothesis. The time spent to produce courseware within these different structures was also estimated.

References

[DeUlloa 1997] J. R. DeUlloa, *The Step by Step Guide to Successfully Promoting a Web Site*, http://www.rankthis.com

[Goldberg 1998] http://www.cranfield.ac.uk/stats/ - Jeff Goldberg, *"Why Web Usage Statistics are (worse than) Meaningless"*

[Linder 1998] http://gopher.nara.gov:70/0h/what/stats/webanal.html - Doug Linder, *"Interpreting WWW Statistics"*

[Maurer 1996] H. Maurer, HyperWave - *the Next Generation Web Solution*, Addison-Wesley, 1996

[Newmarch 1996] J. Newmarch, *Operating Systems Lecture Notes*, http://pandonia.canberra.edu.au/OS/

[Newmarch 1997] J. Newmarch, *Courseware on the Web: an Analysis of Student Use*, AusWeb 1997, Gold Coast, Australia 1997

[Newmarch 1998] J. Newmarch, *An Investigation into the Effect of Hypertext Structure on Student Use of Courseware - long version* http://pandonia.canberra.edu.au/web/usage2/paper-long.html

[Noonan 1998] http://www.piperinfo.com/pl01/usage.html - Dana Noonan, *"Making Sense of Web Usage Statistics"*